(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 830 018 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
18.03.1998 Bulletin 1998/12

(51) Int Cl.6: H04N 5/44

(21) Application number: 97202808.8

(22) Date of filing: 12.09.1997

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

(30) Priority 13.09.1996 US 26026 P

(71) Applicant TEXAS INSTRUMENTS INCORPORATED
Dallas. Texas 75243 (US)

(72) Inventor: Ohara, Kazuhiro
Richardson, Texas 75082 (US)

(74) Representative: Holt, Michael
Texas Instruments Limited,
Kempton Point,
68 Staines Road West
Sunbury-on-Thames, Middlesex TW16 7AX (GB)

(54) Method and system for motion detection in a video image

(57) A method of measuring the motion in video image data for a pixel which uses both field-difference and frame-difference motion values to generate a motion value having increased accuracy. Image data (806) from the same pixel in a prior row of the same field (906) is compared to image data (808) from the same pixel in the prior row of the prior frame (908), and the absolute value of the difference is compared to the absolute value of the difference in image data (802) from the same pixel in a following row of the same field (902) and image data (804) from the same pixel in the following line of the prior frame (904). The minimum of these two values is the minimum frame-difference motion value which is input into a logical mixer. Also input into the logical mixer is the minimum field-difference motion value which may be determined by comparing data (802, 806) from the same pixel of an adjacent line of the same field (902, 906) with image data (810) from the same pixel of the same line of the prior field. The average of image data (810) from the same pixel of the same line of the prior field and image data (812, 814) from the same pixel of two rows prior or two rows after of the prior field (912, 914) may be used instead of image data (810) from the same pixel of the same line of the prior field alone, to increase the accuracy of the measurement.

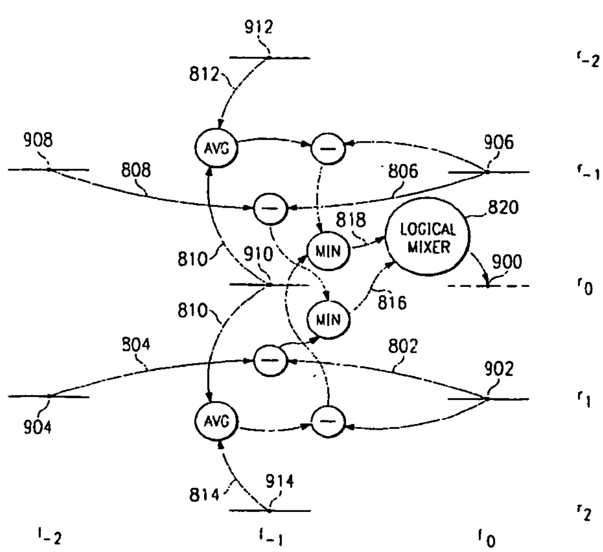


FIG. 9

EP 0 830 018 A2

Printed by Jouve, 75001 PARIS (FR)

Description

FIELD OF THE INVENTION

This invention relates to the field of image display systems, more particularly to methods and systems for detecting motion in a video data sequence.

BACKGROUND OF THE INVENTION

Video image display systems create moving images by rapidly displaying a sequence of still images, or frames. Display systems must rapidly produce each new frame in order to create the impression of smooth motion in the video sequence. Each frame is formed by displaying an orthogonal array of picture elements, or pixels. During each frame, every pixel in the array is assigned its own brightness and color value. Digital systems typically represent a single pixel using three values, each value representing the intensity of one of three component colors.

Due to the high frame rate required to smoothly portray moving objects, display systems require a very high data throughput. Early television broadcast standards, such as NTSC, developed a technique called interlacing to reduce the throughput requirements. Interlaced video systems, such as NTSC, PAL, SECAM, and some HDTV standards, transmit each frame as two sub-frames or fields. Each of the two fields that form a frame contain alternate scan lines from the frame. The first field typically contains all of the odd scan lines while the second field contains all of the even scan lines. Because the display forms the two sequential fields so quickly, the viewer's eye integrates the sequential fields into a continuous moving display. While the two separate fields are visually integrated into a single frame, flicker is reduced by projecting the image fields sequentially.

Modern image display systems, such as most computer displays and some HDTV standards, are non-interlaced. Non-scanned, or simply proscan, since the lines that form each image are scanned sequentially from top to bottom instead of being divided into two fields. Proscan display systems must have a higher frame rate than interlaced systems in order to avoid visible image flicker. Because of the higher frame rate, proscan systems typically display more information and have a higher resolution than comparable interlaced systems with a lower frame rate.

Some modern image display systems with relatively high bandwidths convert interlaced video signals to proscan in order to improve the display quality. Additionally, some display devices, such as the digital micromirror device (DMD), utilize proscan data conversion to compensate for a lack of image persistence.

Proscan conversion can introduce errors, or artifacts, into an image depending on what the video sequence is displaying and how the proscan conversion is being performed. A simple form of proscan conversion simply adds the even lines from a frame to the odd lines of a frame. Although this form of proscan conversion is preferred for still images, it creates problems when displaying moving objects. The problems arise from the fact that the two fields in an image frame do not represent the image at the same point in time. The image data is created by scanning the original image twice, once for every odd line and a second time for every even line, therefore the even-line field represents data one-half of a frame period later than the data represented by the odd-line field. The proscan conversion described above, which creates current frame images by filling in missing lines with pixel data from the prior field, causes misalignment in moving images. This misalignment is most obvious along the edges of a moving object since the edges will appear jagged. The same effect occurs in the center of a moving object, but unless there is a lot of contrast within the object the artifacts are not as noticeable.

Alternative forms of proscan conversion, which eliminate the effects of motion, are line doubling and line averaging. Both line doubling and line averaging use data from adjacent pixels of the same field to fill in the missing lines of the current field. Line doubling simply displays each line from the present field twice, once in its proper position and once in place of the subsequent or preceding line from the next field. When the next field is received, the display again uses each line of image data twice, once in its proper position and once in place of the preceding or subsequent line from the previous field. Line-averaging systems create a new line of image data based on the average of the image data for the lines above and below the created line. Because both the line-doubling and line-averaging methods only use data from one time sample, they avoid the problems associated with simply combining the two image fields. Line-doubling and line-averaging, however, reduce the effective resolution of the image since they use less information to generate each image.

In order to maintain the highest effective resolution while avoiding motion artifacts, proscan conversion systems should compensate for motion in the image data. Ideally, the contribution of adjacent pixels from the same video field and from the same pixel in adjacent video fields should depend on the amount of motion in the video sequence. Therefore an accurate motion detection system is needed to allow optimization of the proscan conversion process.

SUMMARY OF THE INVENTION

Objects and advantages will be obvious, and will in part appear hereinafter and will be accomplished by the present invention which provides an improved method and system for measuring motion in a video image and for performing a proscan conversion on interlaced video data based on the improved motion measurement.

According to a first embodiment of the improved method of measuring motion, a field-difference motion value is measuring motion, a field-difference motion value is calculated for a missing pixel. This field-difference motion value is used to select a proscan algorithm that will accurately de-interlace the video data.

According to another embodiment of the improved method of measuring motion, a field-difference motion value is determined by calculating a first prior-field average value equal to the average of same-pixel prior-field image data and same-pixel prior-field two-rows-prior image data, and determining the absolute value of the difference between same-pixel prior-row image data and the first prior-field average value data.

According to yet another embodiment of the improved method of measuring motion, a motion value is determined by determining a first prior-field average value equal to the average of same-pixel prior-field image data and same-pixel prior-field two-rows-prior image data, determining a first field-difference motion value equal to the absolute value of the difference between same-pixel prior-row image data and the first prior-field average value data.

According to yet another embodiment of the improved method of measuring motion, a motion value is determined by determining a second prior-field average value equal to the average of same-pixel prior-field image data and same-pixel prior-field two-rows-later image data, and determining second field-difference motion value equal to the absolute value of the difference between same-pixel prior-row image data and the second prior-field average value data.

According to yet another embodiment of the improved method of measuring motion, a first field-difference motion value is determined by determining a first prior-field average value equal to the average of same-pixel prior-field image data and same-pixel prior-field two-rows-prior image data, determining a first field-difference motion value equal to the absolute value of the difference between same-pixel prior-row image data and the first prior-field average value data, a second field-difference is determined by determining a second prior-field average value equal to the average of same-pixel prior-field image data and same-pixel prior-field two-rows-later image data, and determining second field-difference motion value equal to the absolute value of the difference between same-pixel prior-row image data and the second prior-field average value data, and a minimum of the first and second field-difference motion values is used as the field-difference motion value.

According to yet another embodiment of the disclosed invention calculates a field-difference motion value for a missing pixel, calculates a frame-difference motion value for a missing pixel, and selects a proscan algorithm based on both the frame-difference and the field-difference motion values to select an algorithm for creating data for the missing pixel.

According to yet another embodiment of the disclosed invention, a proscan algorithm is selected based on the frame-difference motion value when the frame-difference motion value is less than a threshold and using the field-difference motion value when the frame-difference motion value is greater than the threshold.

According to yet another embodiment of the disclosed invention, a proscan algorithm is selected based on the frame-difference motion value when the frame-difference motion value is less than a first threshold, using the field-difference motion value when the frame-difference motion value is greater than a second threshold, and using a weighted average of the frame-difference and the field-difference motion values to select an algorithm for creating data for the missing pixel when the frame-difference motion value is less than the first threshold and greater than the second threshold.

According to yet another embodiment of the disclosed invention, a method of determining a motion value for a pixel location in a video signal is provided. The method comprises determining a first frame-difference motion value by comparing same-pixel prior-row image data from a current frame and a value by comparing same-pixel next-row image data from the current frame and the prior frame, and setting the motion value equal to a minimum of the first frame-difference motion value and the second frame-difference motion value.

According to yet another embodiment of the disclosed invention, a method of determining a motion value is provided. The method comprises comparing same-pixel prior-row image data from a current frame and a same-pixel same-row image data from a prior field.

According to yet another embodiment of the disclosed invention, a logical mixer is provided. The logical mixer comprises a first comparator outputting a selection signal indicating whether a first signal is greater than a threshold signal, a second comparator outputting a maximum signal equal to the maximum of the first signal and a second signal, and a selector receiving the selection signal, the first signal, and the maximum signal, the selector outputting the first signal when the first signal is less than the threshold signal and outputting the maximum signal when the first signal is greater than the threshold signal.

According to yet another embodiment of the disclosed invention, a display system is provided. The display system comprises a video processor for receiving an interlaced video signal and converting the interlaced video signal to a

progressive-scan video signal, the video processor performs the conversion based on a calculated field-difference motion value for the interlaced video signal, and a display for receiving the progressive-scan video signal from the video processor and for displaying the progressive scan video signal.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention, and the advantages thereof, reference is now made to the following descriptions taken in conjunction with the accompanying drawings, in which: accompanying drawings, in which:

Figure 1 is a graphical representation of five lines of image data from each of three fields of a video data sequence;
Figure 2 is a block diagram of a frame-difference motion detection system of the prior art;
Figure 3 is schematic representation of the frame-difference motion detection system of Figure 2 indicating the location of the two pixels used to generate a motion value for a third pixel;
Figure 4 is a block diagram of an improved frame-difference motion detection system which uses data from two rows of two frames to generate a motion value for a pixel;
Figure 5 is a schematic representation of the improved frame-difference motion detection system of Figure 4 indicating the location of the four pixels used to generate a motion value for a fifth pixel;
Figure 6 is a block diagram of a field-difference motion detection system which uses same-line-previous-field data and previous-line-same-field data to generate a motion value for a pixel;
Figure 7 is a schematic representation of the field-difference motion detection system of Figure 6 indicating the location of the two pixels used to generate a motion value for a third pixel;
Figure 8 is a block diagram of an improved motion detection system which uses both field and frame-difference techniques to generate a motion value for a pixel based on data from five rows and three frames of pixels;
Figure 9 is a schematic representation of the improved motion detection system of Figure 8 indicating the location of the seven pixels used to generate a motion value for an eighth pixel;
Figure 10 is a block diagram of one embodiment of the logical mixer of Figure 8 used to combine the results from a field-difference motion detection subsystem and a frame-difference motion detection subsystem;
Figure 11 is a block diagram of a second embodiment of the logical mixer of Figure 8 used to combine the results from a field-difference motion detection subsystem and a frame-difference motion detection subsystem;
Figure 12 is a block diagram of a third embodiment of the logical mixer of Figure 8 that performs a soft-switching function by gradually shifting from a frame-difference output to a field-difference output; and
Figure 13 is a block diagram of a display system having a video processor that uses both frame-difference and field-difference motion detection.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figure 1 schematically depicts video data for five adjacent image lines from each of three sequential fields. Each line of real image data, image data transmitted from the interleaved source, is designated by a solid line. Each line of created image data, data generated by the proscan conversion, is designated by a dashed line. The spatial relationship between image lines within a single field is shown in Figure 1 which shows the five rows, r_{-2}, r_{-1}, r_0, r_1, and r_2, arranged vertically adjacent to each other. The current line, the line for which motion vectors are presently being calculated, is row r_0. Three sequential fields, f_{-2}, f_{-1}, and f_0, are shown in chronological order from left to right. The current field, the field for which motion vectors are presently being calculated, is f_0. As shown in Figure 1, field f_{-2} is comprised of the odd-numbered rows, r_{-1} and r_1, of a first frame, field f_{-1} is comprised of even-numbered rows r_{-2}, r_0, and r_2, and field f_0 is comprised of the odd-numbered rows, r_{-1} and r_1, from a second frame.

The dark circle on each of the image lines represents a single pixel in the row of pixels. Unless otherwise specified, the pixel indicated in each row of a figure is the same pixel from each row, i.e. the pixel occupying the same horizontal position in each of the rows indicated. proscan conversion to create image data for pixel **102** in row r_0 of field f_0 from data for pixels **104, 108**, and **110** of an interlaced video signal. Typical motion detection algorithms subtract the pixel data of pixel **104** from the pixel data of pixel **106** and use the magnitude of the difference between the two pixels as a rough estimate of the amount of motion in the portion of the image depicted by pixel **102.** If no motion is detected, the proscan conversion algorithm simply uses the data from pixel **108** for pixel **102.** If a lot of motion is detected, the average of the image data from pixel **104** and pixel **110** is used for pixel **102.** For intermediate motion values, a weighted average of the data from each of the three pixels is used with the weights selected to transition smoothly from the no motion use of data from pixel **108** to the high motion use of data from pixel **104** and pixel **110.** For example, the image data for pixel **102**, ID_{102}, is determined by:

$$ID_{102} = (k^*ID_{108}) + (1\text{-}k)^*(ID_{104}+ID_{110})/2$$

where ID_x is the image data for pixel x, and k is the magnitude of the detected motion ranging from zero to one. Since there is no need to generate image data for the image lines which are transmitted in each field, the motion detection algorithm is only used on alternate lines within each field.

Figure 2 is a block diagram of a motion detection system **200** according to the prior art. In Figure 2, delay block **202** delays the current image data **204** for one field period. Summation block **206** subtracts the delayed image data **208** from the current image data **204.** The output of the summation block **206** is a number representing the detected motion at the current image pixel. Block **210** determines the absolute value of the number representing the detected motion and block **212** scales the absolute value. Non-linear scaling block **212** typically subtracts a minimum threshold value from the absolute value and limits, or clips, the result to a maximum value. The minimum threshold provides some immunity to noise on the motion signal while the maximum value narrows the range of motion values that must be considered by the proscan of motion values that must be considered by the proscan algorithm.

In practice, the minimum threshold values determine the point at which the proscan conversion algorithm starts using current field data to generate image data, the point at which k is greater than zero in the previous equation, while the limiting value determines the point at which the proscan conversion algorithm stops using data from a previous field to generate image data, k equals 1. Therefore, the frame-difference algorithm of Figure 2 uses current field and previous frame data to determine a motion value, k, which controls how image data is created from current field and previous field data.

Figure 3 depicts the operation of the frame-difference algorithm shown in Figure 2. In Figure 3, motion data for pixel **302** is generated by subtracting the prior frame adjacent pixel **306** (f_{-2}, r_{-1}) data from the current frame adjacent pixel **304** (f_0, r_{-1}) data. Not shown in Figure 3 are the absolute value or the non-linear scaling functions of Figure 2. Depending on the motion value determined for pixel **302**, image data for pixel **302** is created using image data from pixels **304, 308,** and **310**.

An improved frame-difference system and algorithm, which uses data from adjacent rows above and below the missing data line is depicted in Figures 4 and 5. In Figures 4 and 5, the image data for row r_0 is generated using data from rows r_{-1} and r_1. In Figure 4, the current image data **402** is compared to image data from the previous frame **404**, and image data from the previous line **406** is compared to image data from the previous line of the previous frame **408.** Magnitude scaling blocks **410,** and **412,** scale the absolute value of the results of these comparisons resulting in a motion value **414** for the line below the missing line and a motion value **416** for the line above the missing line. These two motion values are compared and the minimum motion value **418** is used to influence the proscan algorithm. Instead of merely selecting the to select the maximum motion value, or a weighted average of the two calculated motion values **414** and **416**. Figure 5 graphically shows the data flow through the system of Figure 4. The absolute value/non-linear scaling functional blocks, **410** and **412,** of Figure 4 are not shown in Figure 5.

The frame-difference systems and algorithms represented by Figures 2-5 reliably measure the image motion in still, or low-motion video sequences. They may, however, fail to detect motion in very high-motion image sequences due to the relatively slow sample rate of the input data.

An alternative to the frame-difference systems and algorithms of Figure 2-5 is the field-difference motion detection system **600** of Figure 6. In Figure 6, the frame delay of Figure 2 has been replaced by a field delay **602.** Referring to Figure 7, the field-difference system **600** of Figure 6 compares transmitted image data from pixels **702** (f_0, r_{-1}) and **704** (f_{-1}, r_0) to generate a detected motion value for pixel **706** (f_0, r_0). As in the frame-detection system of Figure 2, the magnitude of the detect motion value is scaled by non-linear scaling block **212** before it is used to control the proscan conversion algorithm.

The advantage of the field-difference motion detection system **600** of Figure 6 is its enhanced ability to detect rapid motion compared to the detection system of Figure 2. The enhanced detection ability is due to the increased sample rate of the field-difference motion detection system **600** compared to the frame sample rate used by the detection system **200** of Figure 2. The field-difference motion detection system **600** uses current field (f_0) and previous field (f_{-1}) data instead of current frame and previous frame data to determine a motion value which controls how image data is created from current field and previous field data.

Unfortunately, while the field-difference motion detection system **600** detects very high rates of motion, it also generates false motion values when the processed image has a high vertical image frequency since the two data values has a high vertical image frequency since the two data values being compared are from different rows of the image. The image data in each line in an image with a high vertical image frequency is very different from the image data in adjacent lines. Therefore, when the field-difference motion detection system **600** compares data from pixel **702** with data from pixel **704,** the line-to-line difference in still image data is interpreted as motion.

To overcome the problems inherent with frame-difference and field-difference motion detection systems, a new detection system has been developed. The disclosed motion detection system **800** is shown in Figure 8. Figure 9

depicts the logical operation of the motion detection system **800** on image data. The new motion detection system of Figures 8 and 9 determines the motion vector for each pixel in each of the missing rows by measuring and comparing four different motion vectors. First, the system of Figures 8 and 9 measures the frame-difference motion vector for the same pixel, the pixel in the same horizontal location, in the lines immediately above and below the pixel of interest. Then, the system measures the field-difference motion vector for the same two pixels. The lesser of the two frame-difference motion vectors and the lesser of the two field-difference motion vectors are input into a logical mixer and used to determine the motion vector to be assigned to the current pixel.

Figures 8 and 9 show the steps used to calculate these four motion vectors. As shown in Figure 8, one frame-difference motion value is computed by comparing the current image data **802** from pixel **902** (f_0, r_1) of Figure 9 with image data **804** from the same pixel of the previous frame **904** (f_{-2}, r_1) using block **805** while the second frame-difference motion value is computed using image data **808** from the same pixel of the previous line **906** (f_0, r_{-1}) and image data **810** from the same pixel of the previous line of the previous frame **908** (f_{-2}, r_{-1}) by block **809.** Comparator **817** outputs the minimum value of these two comparisons as the frame-difference motion value **816** of Figure 4.

The motion detection system of Figure 8 also calculates a minimum field-difference motion value **818** by averaging the image data from pixels **910** (f_{-1}, r_0) and **912** (f_{-1}, r_{-2}) and comparing the average to the image data value from pixel **906** (f_0, r_{-1}). The result is compared with a second field-difference value determined using data from pixels **902** (f_0, r_1), 910 (f_{-1}, r_0), and **914** (f_{-1}, r_2), and the minimum of the two operations is the minimum field-difference value **818.**

Although Figures 8 and 9 each show selecting the minimum field-difference motion value **818** and the minimum frame-difference motion value **814** of the two values created, the motion detection system **800** could also select the maximum motion value, the average motion value, or a weighted average of the two motion value depending on the application.

The minimum frame-difference motion value **816** and the minimum field-difference motion value **818** are both input into a logical mixer **820.** The logical mixer **820** outputs a motion value which is then smoothed by an optional vertical maximum detector **822** and an optional vertical/horizontal low pass filter **824** before being scaled by an optional non-linear scaling function **826.** The vertical maximum detector **822** compares the motion value of the current line with the motion value for a pixel in the same location of the previous, current, and next lines, and uses the maximum of the three values as the current pixel's motion value. The vertical maximum detector **822** and the vertical/horizontal low pass filter **824** both operate to smooth the output of the motion detection system **800.** This smoothing helps to prevent visual artifacts from being introduced between regions in which motion is detected and those in which no motion is detected. Additionally, a non-linear scaling block **826,** which typically subtracts a minimum threshold value from the motion value and limits, or clips, the result to a maximum value, also operates to smooth the motion value determined by the circuit of Figure 8.

The disclosed motion detection circuit relies on a new logical mixer **820** to determine the proper motion value. The logical mixer **820** is critical to obtaining the accurate motion data from the field and frame-difference portions of the motion detection system **800.** Because the frame-difference portion of the motion detection system **800** is very accurate for low frequency motion, the logical mixer block outputs the results of the frame-difference subcircuit when the detected motion value is low. When the detected motion value is high, however, the logical mixer block outputs the results of the field-difference subcircuit.

To summarize Figures 8 and 9, a motion value for the current pixel **900** is determined by first finding a field-difference motion value **818** and a frame-difference motion value **816** and inputting the field-difference motion value **818** and frame-difference motion value **816** into logical mixer **820.** The frame-difference motion value is the minimum of two frame-difference motion values obtained using adjacent data from the same field (f_0) and the previous frame (f_{-2}). The first frame-difference motion value is the absolute value of the difference between image data **806** from the same pixel **906** in a prior row (r_{-1}) of the same field (f_0) and image data **808** from the same pixel **908** in the prior row (r_{-1}) of the prior frame (f_{-2}). The second frame-difference motion value is the absolute value of the difference between image data **802** from the same pixel **902** in a following row (r_1) of the same field (f_0) and image data **804** from the same pixel **904** in the following row (r_1) of the prior frame (f_{-2}). In other words, the frame-difference motion value is the minimum absolute value obtained by comparing same-pixel adjacent-row same-field image data **802, 806,** with same-pixel adjacent-row prior-frame image data **804, 808.**

Also input into the logical mixer is the field-difference motion value. The field-difference motion value is the minimum of two field-difference motion values obtained using adjacent data from the same field (f_0) and the previous field (f_{-1}). The first field-difference motion value is the absolute value of the difference between image data **806** from the same pixel **906** of the prior row (r_{-1}) of the same field (f_0) and the average of image data **810** from the same pixel **910** of the same row (r_0) of the prior field (f_{-1}) and image data **812** from the same pixel **912** of two rows prior (r_{-2}) in the prior field (f_{-1}). The second field-difference motion value is the absolute value of the difference between image data **802** from the same pixel **902** of the following row (r_1) of the same field (f_0) and the average of image data **810** from the same pixel **910** of the same row (r_0) of the prior field (f_{-1}) and image data **814** from the same pixel **914** of two rows following (r_2) in the prior field (f_{-1}). In other words, the field-difference motion value is the minimum absolute value obtained by

comparing same-pixel adjacent-row same-field image data **802, 806,** with the average of same-pixel same-row prior-field image data **810** and same-pixel two-rows-adjacent prior-field image data **812, 814.** Combining the teachings of Figures 6 and 7 with that of Figures 8 and 9, image data **810** could be substituted for the average of image data **810** and the data **812, 814,** from the same pixels **912, 914,** two rows adjacent (r_{-2}, r_2) in the prior field (f_{-1}).

Figure 10 shows one embodiment of a logical mixer according to the present invention. In Figure 10, the minimum frame-difference value (MFR) is compared to a threshold value (TH) by comparator **1002.** If the minimum frame-difference value is greater than the threshold, then the greater of the frame-difference motion value and the field-difference motion value is output by selector **1004.** The logical mixer **1000** of Figure 10 selects the frame-difference motion value when there is only a small amount of motion in the image and the greater of the frame and field-difference motion values when there is a lot of motion in the image. The result is that the frame-difference motion value, which is very accurate, is used except when both the field-difference value is greater and the frame-difference value is significant. This allows the faster field-difference value to be used without the fear of falsely detecting motion in still images.

Figure 11 shows a second embodiment of a logical mixer where the threshold value is three. In Figure 11, whenever either one or both of the two most significant bits, MFR[3] or MFR[2], is set, the maximum of the minimum frame difference (MFR[3:0]) and minimum field difference (MFI[3:0]) motion values is used by the display system to calculate a current pixel motion value.

Figure 12 shows an embodiment of a logical mixer **1200** that implements a soft switching function. In Figure 12, comparator **1202** compares the minimum frame difference motion value (MFR[3:0]) to a lower threshold (THA), an intermediate threshold (THB), and an upper threshold (THC), and outputs a signal to multiplexer **1204** to indicate which thresholds, if any, the minimum frame difference motion value exceeds. The minimum frame-difference motion value is output by selector **1202** whenever it is less than a lower threshold (THA), and the maximum of the field and frame-difference values is used when the minimum frame-difference motion value is above an upper threshold (THC). When the minimum frame-difference value is above the lower threshold, but below the upper threshold (THC), a weighted average of the minimum frame-difference and the minimum field-difference values, determined by averaging circuitry **1206,** is used. In the example shown in Figure 12, an intermediate threshold (THB) is used to allow one of four combinations of the frame and field-difference values to control the proscan algorithm.

Figure 13 shows one embodiment of a display system including the disclosed motion detection system. In Figure 13, video data is received by receiver **1302.** The video processor **1304** includes the disclosed motion detection system and uses the disclosed motion detection system to process the received video data. The processed video data is then output to display device **1306** for display.

Although to this point the disclosed motion detection Although to this point the disclosed motion detection system has been described in terms of block diagrams implementing the system, it should be understood that the system can be implemented in many ways, including hardware and software components. Table 1 below is a listing of a software program implementing the disclosed motion detection system.

Table 1

```
;*****************************************************************
*********
;*
;*      CODE FOR MOTION DETECTION COMPARISON DEMO
;*              (ONE ALGORITHM ON ONE SCREEN)
;*   This code is for "Motion Detection using Field and Frame
;*   Difference, up to Logical Mixer" algorithm
;*
;*      lib         : mac0_310.lib
;*      assembler : asm_404.exe
;*
;*      Imodes     :
;*                Im  =  1  :  Motion  Detection  using  Field/Frame
Difference      *
;*          Im = 2 : Motion Detection using Field Difference
;*          Im = 3 : Motion Detection using Frame Difference
;*          Im = 4 : Pass through
```

```
;*          Im = 5 : Pass through

;*          Im = 6 : Pass through

;*        * Im = 7 : Left Output = 2, Right Output = 1

;*        * Im = 8 : Left Output = 3, Right Output = 1

;*        * Im = 9 : Left Output = 1, Right Output = 2

;*        * Im =10 : Left Output = 1, Right Output = 3

;*        * Im =11 : Left Output = 1, Right Output = 4

;*        * Im =12 : Left Output = 1, Right Output = 5

;*        * Im =13 : Left Output = 1, Right Output = 6

;*        * Im =14 : Left Output = 6, Right Output = 4

;*        * Im =15 : Left Output = 5, Right Output = 4

;*

;*    Imodes 7 through 15 are not available with this program.

;*

;***************************************************************

     ERI
     .width    132
       .mlib      c:\svp\lib\mac0_310.lib
;        .mlib      c:\lib\mac0_310.lib
```

```
;**********    RF Regions    **********

RF0:
;---------------------
VM00    .set0    0          ; 0H delayed Logical
Mixer Output
VL0C    .set0    4          ; 0H delayed Vertical
Max Function Output
VM01    .set0    8          ; 1H delayed Logical
Mixer Output
VL01    .set0    12         ; 1H delayed Vertical
Max Function Output
VM02    .set0    16         ; 2H delayed Logical
Mixer Output
VL02    .set0    20         ; 2H delayed Vertical
Max Function Output
VM03    .set0    24         ; 3H delayed Logical
Mixer Output
VL03    .set0    28         ; 3H delayed Vertical
Max Function Output
VM04    .set0    32         ; 4H delayed Logical
Mixer Output
VL04    .set0    36         ; 4H delayed Vertical
Max Function Output
D01     .set0    50         ; Scratch memory
D02     .set0    60         ; Scratch memory
D03     .set0    70         ; Scratch memory
L00     .set0    81         ; Scratch memory
L01     .set0    91         ; Scratch memory
F0      .set0    127        ; Scratch memory

RF1:
;---------------------
E10     .set1    0          ; 262H delayed
Luminance
E11     .set1    8          ; 263H delayed
```

```
Luminance
E12         .setl        16                              ; 264H delayed
Luminance
F10         .setl        24                              ; 524H delayed
Luminance
F11         .setl        32                              ; 525H delayed
Luminance
F12         .setl        40                              ; 526H delayed
Luminance
D10       .setl      48                        ; Luminance
D11         .setl        56                                ; 1H delayed
Luminance
D12         .setl        64                                ; 2H delayed
Luminance
L10       .setl      72                        ; Scratch memory
L11       .setl      76                        ; Scratch memory
L12       .setl      80                        ; Scratch memory
F1        .setl     127                        ; Scratch memory

;         DOR
;---------------------
MM0        .set         8                          ; Motion Magnitude
Output
YOUT      .set       0

;         DIR
;---------------------
Y         .set       0                         ; Luminance
Y262       .set         32                           ; 262H delayed
Luminance
Y524       .set         8                            ; 524H delayed
Luminance

      lrm0        40, 8
      lrm1        72, 8
```

```
;**********    IG Code    **********

;********************                    HSYNC      1
*************************************
hsyncl: jfaz      $                        ; Wait for Flag A

     mov          out'MM0,  R0'L00+1'4     ; output Left half
Motion Magnitude
     mov          out'YOUT,  R0'D03'8      ; output Left half
Luminance

     grl0                                  ; Global Rotate RF0
     grl1                                  ; Global Rotate RF1

     umr                                   ; Update Mode Resister

     mov        R1'D10, inp'Y'8            ; input Luminance
     mov          R1'E10,  inp'Y262'8      ; input 262H delayed
Luminance
     mov          R1'F10,  inp'Y524'8      ; input 524H delayed
Luminance

;          jme        lum, 2

     mov          R0'D03,  R1'D12'8        ; D03 = 2H delayed
Luminance
;          jmp        next

;lum:     mov         R0'D03, R1'D10'8
                              ; Pass Through Mode
next:    jme         hsyncl,  4            ; if Imode=4, then
jump to hsyncl
     jme        hsyncl, 5                  ; if Imode=5, then jump
to hsyncl
     jme        hsyncl, 6                  ; if Imode=6, then jump
to hsyncl
```

```
        mov        R0'L00, R1'F11'8          ; L00 = 525H delayed
Luminance
        mov        R0'L01, R1'F12'8          ; L01 = 526H delayed
Luminance
        mov        R0'D01, R1'E11'8          ; D01 = 263H delayed
Luminance
        mov        R0'D02, R1'E12'8          ; D01 = 264H delayed
Luminance

                                    ; Motion Detection using Field
Difference
        jme        mfi, 2                    ; if Imode = 2, then
jump to MFI

;******************                                          MFR
*******************************************************
;******************              Frame          Difference       1
*********************************
;******************                     Subtraction               1
**************************************
;*
        *
;*   Subtract from Previous Frame
        *
;*      ( 525H delayed Luminance - Luminance )
        *
;*
        *
;*****************************************************************
********
mfr:    sub        R0'L00, R0'L00'8, R1'D10'8, 0
                                ; L00 = 525H delayed Luma - Luma

;******************   Absolute  /  Non-Linear  Function  1
******************
```

```
;*
        *
;*  Absolute Previous result, and add -4
        *
;*     ( ABS(L00) + (-4) )
        *
;*
        *
;*************************************************************
*********
     abs       R0'L00, R0'L00'9, 1     ; L00 = ABS(L00)
     addc      R0'L00, -4'4, R0'L00'9, 1
                          ; L00 = L00 + (-4)
     clpzt     R0'L00'4, R0'L00'10, R1'F1
                          ; Clip L00(11 bits) to (4 bits) and
save

;******************         Frame          Difference           2
*********************************
;******************                Subtraction                  2
*************************************
;*
        *
;*  Subtract from 1H delayed Previous Frame
        *
;*     ( 526H delayed Luminance - 1H delayed Luminance )
        *
;*
        *
;*************************************************************
*********
     sub       R0'L01, R0'L01'8, R1'D11'8, 0
                          ; L01 = 526H delayed Luma - 1H
delayed Luma

;******************  Absolute   /   Non-Linear   Function   2
```

```
******************
;*
        *
;*  Absolute Previous result, and add -4
        *
;*    ( ABS(L01) + (-4) )
        *
;*
        *
;***************************************************************
*********
     abs        R0'L01, R0'L01'9, 1     ; L01 = ABS(L01)
     addc       R0'L01, -4'4, R0'L01'9, 1
                            ; L01 = L01 + (-4)
     clpzt      R1'L11'4, R0'L01'10, R1'F1
                            ; Clip L01(11 bits) to (4 bits)
                            ;        and save to L11
;******************       Min       Funciton      for      MFR
******************************
;*
        *
;*  Take the smaller of Frame Difference 1 and 2
        *
;*    ( MIN(L00, L11) )
        *
;*
        *
;***************************************************************
*********
     min        R0'L00, R0'L00, R1'L11'4, R0'F0, 0
                            ; L00 = MIN(L00, L11)

                             ;  Motion  Detection  using  Frame
Difference
     jme        move, 3                  ; if Imode=3, then jump
to MOVE
```

```
;******************                                              MFI
*********************************************
;******************              Field         Difference          1
*******************************
;******************               Line          Average            1
**********************************
;*
        *
;*   Take  line  average  of  262H  delayed  Luminance  and  263H
delayed      *
;*  Luminance, and subtract from present Field Luminance
        *
;*     ( Luminance - 1/2 * (262H delayed Luma + 263H delayed
Luma) )     *
;*
        *
;************************************************************
*********
mfi:     add       R0'D01, R0'D01'8, R1'E10'8, 0
                                 ; D01  =  262H  delayed  Luma  +  263H
delayed Luma
                                 ; D01 = (1/2) * D01
     sub       R0'D01+1, R1'D10'8, R0'D01+1'8, 0
                                 ; D01 = (1/2)*D01 - Luminance

;******************    Absolute   /   Non-linear   Function   1
******************
;*
        *
;*  Absolute Previous result, and add -4
        *
;*    ( ABS(D01) + (-4) )
        *
;*
        *
```

```
;***********************************************************************
*********
      abs         R0'D01+1, R0'D01+1'9, 0; D01 = ABS(D01)
      addc        R0'D01+1, -4'4, R0'D01+1'8, 1
                                ; D01 = D01 + (-4)
      clpzt       R0'D01+1'4, R0'D01+1'9, R1'F1
                                ; Clip D01(11 bits) to (4 bits) and
save

;******************          Field          Difference          2
*********************************
;******************           Line          Average             2
*************************************
;*
        *
;*   Take line average of 263H delayed Luminance 264H delayed
        *
;*   Luminance, and subtract from 1H delayed Field Luminance
        *
;*     ( 1H delayed Luminance -
        *
;*                  (263H delayed Luma + 264H delayed Luma)
        *
;*
        *
;***********************************************************************
*********
      add         R0'D02, R0'D02'8, R1'E11'8, 0
                                ; D02 = 263H delayed Luma + 264H
delayed Luma
                                ; D02 = (1/2) * D02
      sub         R0'D02+1, R1'D11'8, R0'D02+1'8, 0
                                ; D02 = (1/2) * D02 - 1H delayed
Luma

;******************   Absolute  /  Non-linear   Function   2
```

```
******************
;*
        *
;*  Absolute Previous result, and add -4
        *
;*     ( ABS(D02) + (-4) )
        *
;*
        *
;*****************************************************************
*********
     abs            R0'D02+1,  R0'D02+1'9,  0;  D02  =  ABS(D02)
extending 1 bit
     addc      R0'D02+1, -4'4, R0'D02+1'8, 1
                              ; D01 = D01 + (-4)
     clpzt     R1'L12'4, R0'D02+1'9, R1'F1
                              ; Clip D02(11 bits) to (4 bits)
                              ;         and save to L12
;******************      Min      Function       for       MFI
*****************************
;*
        *
;*  Take the smaller of Field Difference 1 and 2
        *
;*     ( MIN(D01+1, L12) )
        *
;*
        *
;*****************************************************************
*********
     min       R1'L12, R1'L12, R0'D01+1'4, R1'F1, 0
                              ; L12 = MIN(L12, D01)

                              ;  Motion  Detection  using  Field
Difference
     jme       max3, 2                    ; if Imode=2, then jump
```

```
to MAX3

;******************               Logical               Mixer
*************************************
lmix:   mov     R1'L10, R0'L00'4        ; L10 = MFR

;******************          Create        Select        Signal
******************************
;*
        *
;*  By taking NOR of MSB and 2nd MSB, create Select Signal
        *
;*    ( NOR(MSB of MFR, 2nd MSB of MFR) )
        *
;*
        *
;*****************************************************************
*********
     or       R1'L10+3, R1'L10+3, R0'L00+2'1
                              ; L10+3 = OR(MSB of MFR, 2nd MSB of
MFR)
     not      R1'L10+3, R1'L10+3'1    ; L10+3 = NOT(L10+3)

;******************               Max                 Function
**************************************
;*
        *
;*  Take max of MFR and MFI
        *
;*    ( MAX (MFR, MFI) )
        *
;*
        *
;*****************************************************************
*********
     max      R1'L12, R1'L12, R0'L00'4, R1'F1, 0
```

```
                                    ; L12 = Max(MFR, MFI)

;******************                                               Selector
***************************************
;*
        *
;*   Select  MFR  or  Max  of  MFR  and  MFI,  depending  on  Select
Signal       *
;*
        *
;*************************************************************
*********
     kmov      R1'L12, R0'L00'4, R1'L10+3
                                    ;  Select  :  if  L10+3  =  1,  L12  =
L00(MFR)
                                    ;          if L10+3 = 0, L12 = L12

     jmp       max3                 ; Jump to MAX3

;******************          3        Line        Max        Function
*******************************
move:   mov       R1'L12, R0'L00'4        ; L12 = L00 = MFR Min
Function Output

max3:    mov         R0'VM00, R1'L12'4         ;  VM00  =  L12  =
Logical Mixer Output(LMO)
     max       R1'L12, R1'L12, R0'VM01'4, R0'F0, 0
                                    ; L12 = MAX(VM00, 1H delayed LMO)
     max       R1'L12, R1'L12, R0'VM02'4, R0'F0, 0
                                    ; L12 = MAX(L12, 2H delayed LMO)

;******************     5     Line     Vertical     Max     Function
*********************
;        max       R1'L12, R1'L12, R0'VM03'4, R0'F0, 0
                                    ; L12 = MAX(L12, 3H delayed LMO)
;        max       R1'L12, R1'L12, R0'VM04'4, R0'F0, 0
```

```
                        ; L12 = MAX(L12, 4H delayed LMO)

;******************     Vertical     Low     Pass     Filter
**************************
;*
        *
;*  1/4 T^-2 + 1/2 T^-1 + 1/2 + 1/2 T^1 + 1/4 T^2
        *
;*
        *
;*************************************************************
*********
    mov     R0'VL00, R1'L12'4       ; VL00 = L12
    add     R1'L12, R1'L12'4, R0'VL04'4, 0
                        ; L12 = VL00 + 4H delayed VL00
                        ; L12 = 1/2 * L12
    add     R1'L12+1, R1'L12+1'4, R0'VL01'4, 0
                        ; L12 = L12 + 1H delayed VL00
    add     R1'L12+1, R1'L12+1'5, R0'VL02'4, 0
                        ; L12 = L12 + 2H delayed VL00
    add     R1'L12+1, R1'L12+1'6, R0'VL03'4, 0
                        ; L12 = L12 + 3H delayed VL00

;******************     Horizontal     Low     Pass     Filter
************************
;*
        *
;*  H(z) = (z^-1 + z^1)(z^-2 + z^2)(z^-1 + 1)(z^1 + 1) * 1/8
        *
;*
        *
;*************************************************************
*********
    add     R0'L00-3, LR1'L12+2'6, RR1'L12+2'6, 0
                        ; L00 = Left(L12) + Right(L12)
    add     R0'L00-3, L2R0'L00-3'7, R2R0'L00-3'7, 0
```

```
                        ; L00 = 2Left(L00) + 2Right(L00)
     add      R0'L00-3, LR0'L00-3'8, R0'L00-3'8, 0
                        ; L00 = Left(L00) + L00
     add      R0'L00-3, RR0'L00-3,9, R0'L00-3'9, 0
                        ; L00 = Right(L00) + L00

;*****************            Non-linear              Function
*******************************
;*
      *
;*  add -2 to previous result and divide by 2
      *
;*
      *
;*************************************************************
*********
     set      R0'L00+7, 0,1         ; sign L00
     addc     R0'L00, -2,3, R0'L00'8, 1
                        ; L00 = L00 + (-2)
                        ; L00 = 1/2 * L00
     clpzt    R0'L00+1'4, R0'L00+1'8, R1'F1
                        ; Clip L00(8 bits) to (4 bits) and
save

     jmp       hsyncl                    ; Jump and wait for
hsyncl

     .end
;*******************                          EOF
*****************************************
```

Thus, although there has been disclosed to this point a particular embodiment for a motion detection system and a method thereof, it is not intended that such specific references be considered as limitations upon the scope of this invention except in-so-far as set forth in the following claims. Furthermore, having described the invention in connection with certain specific embodiments thereof, it is to be understood that further modifications may now suggest themselves to those skilled in the art, for example, later data fields, if available, could also be used to determine the motion of the image data, it is intended to cover all such modifications as fall within the scope of the appended claims.

Claims

1. A method for converting interlaced video data to progressive scanned data, comprising:

calculating a field-difference motion value for a missing pixel; and
selecting a proscan algorithm based on said field-difference motion value.

2. The method of Claim 1, wherein said step of calculating a field-difference motion value comprising:

determining the absolute value of the difference between same-pixel adjacent-row same-field image data and same-pixel same-row prior-frame image data.

3. The method of Claim 1, wherein said step of calculating a field-difference motion value comprising:

determining a first field-difference motion value equal to the absolute value of the difference between same-pixel prior-row image data and same pixel prior-frame image data; and
determining a second field-difference motion value equal to the absolute value of the difference between same-pixel following-row image data and same pixel prior-frame image data.

4. The method of Claim 3, further comprising the step of:

selecting a minimum of said first and second field-difference motion values as said field-difference motion value.

5. The method of Claim 2, wherein said step of calculating a field-difference motion value comprising:

determining a prior-field average value equal to the average of same-pixel same-row prior-field image data and same-pixel two-rows-adjacent prior-field image data; and
determining the absolute value of the difference between same-pixel adjacent-row image data and said prior-field average value data.

6. The method of any preceding Claim, further comprising the step of:

subtracting a minimum threshold value from said field-difference motion value.

7. The method of any preceding Claim, further comprising the step of:

limiting said field-difference motion value to a maximum value.

8. The method of any preceding Claim, said calculating step comprising:

calculating a first field-difference motion value for a missing pixel;
calculating a second field-difference motion value for a pixel one row after said missing pixel; and
selecting a maximum of said first, second, and a third field difference motion value as said field-difference motion value for said missing pixel.

9. The method of any of Claims 2 to 8, wherein said step of calculating a field-difference motion value comprising:

determining a first prior-field average value equal to the average of same-pixel prior-field image data and same-pixel prior-field two-rows-latter image data; and
determining second field-difference motion value equal to the absolute value of the difference between same-pixel prior-row image data and said second prior-field average value data.

10. The method of Claim 9, further comprising the step of:

selecting a minimum of said first and second field-difference motion values as said field-difference motion value.

11. The method of any preceding Claim, further comprising the step of:

using both said frame-difference and said field-difference motion values to select an algorithm for creating data for said missing pixel.

12. The method of Claim 11, wherein said step of calculating a frame-difference motion value comprising:

calculating a first frame-difference motion value equal to the absolute value of the difference between same-pixel following-row image data and prior-row prior-frame image data.

13. The method of Claim 12, further comprising the step of:

selecting a minimum of said first and second frame-difference motion values as said frame-difference motion value.

14. The method of Claim 12 or Claim 13, further comprising the step of:

comparing said frame-difference motion value to a first threshold;
said step of selecting a proscan algorithm based on said field-difference motion value step comprising the step of using said frame-difference motion value to select an algorithm for creating data for said missing pixel when said frame-difference motion value is less than said first threshold and using said field-difference motion value to select an algorithm for creating data for said missing pixel when said frame-difference motion value is greater than said first threshold.

15. The method of any of Claims 12 to 14, wherein said step of selecting further comprising:

comparing said frame-difference motion value to a first threshold;
comparing said frame-difference motion value to a second threshold;
said selecting a proscan algorithm based on said field-difference motion value step further comprising:
using said frame-difference motion value to select an algorithm for creating data for said missing pixel when said frame-difference motion value is less than said first threshold;
using said field-difference motion value to select an algorithm for creating data for said missing pixel when said frame-difference motion value is greater than said second threshold.
Using a weighted average of said frame-difference and said field-difference motion values to select an algorithm for creating data for said missing pixel when said frame-difference motion value is less than said first threshold and greater than said second threshold.

16. A logical mixer comprising:

a first comparator outputting a selection signal indicating whether a first signal is greater than a threshold signal;
a second comparator outputting a maximum signal equal to the maximum of said first signal and a second signal; and
a selector receiving said selection signal, said first signal, and said maximum signal, said selector outputting said first signal when said first signal is less than said threshold signal and outputting said maximum signal when said first signal is greater than said threshold signal.

17. The mixer of Claim 16, wherein a first threshold motion value and a second threshold motion value are providing the same threshold motion value.

18. The mixer of Claim 16 or Claim 17, wherein said motion value for said pixel location is equated to a weighted average of said minimum frame-difference motion value and said maximum motion value when said minimum frame-difference motion value is greater than said first threshold motion value and less than said second threshold motion value.

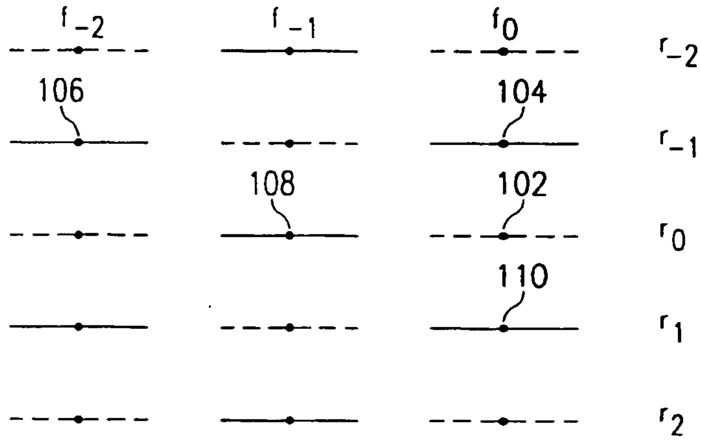


FIG. 1
(PRIOR ART)

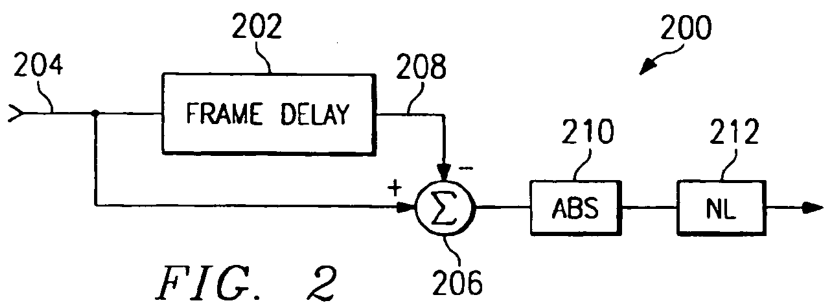


FIG. 2
(PRIOR ART)

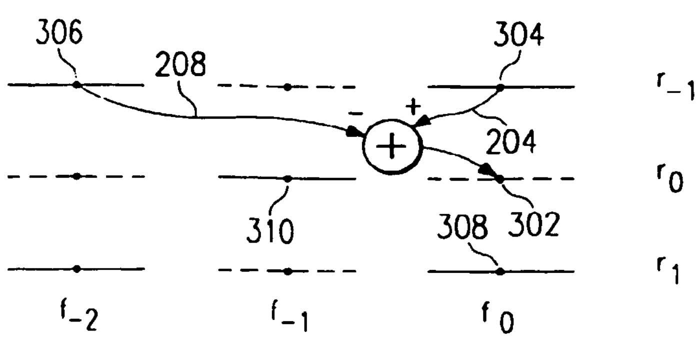


FIG. 3
(PRIOR ART)

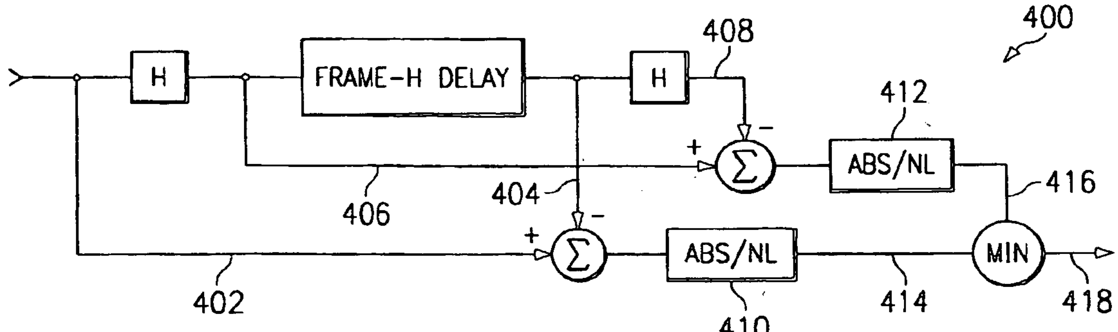


FIG. 4

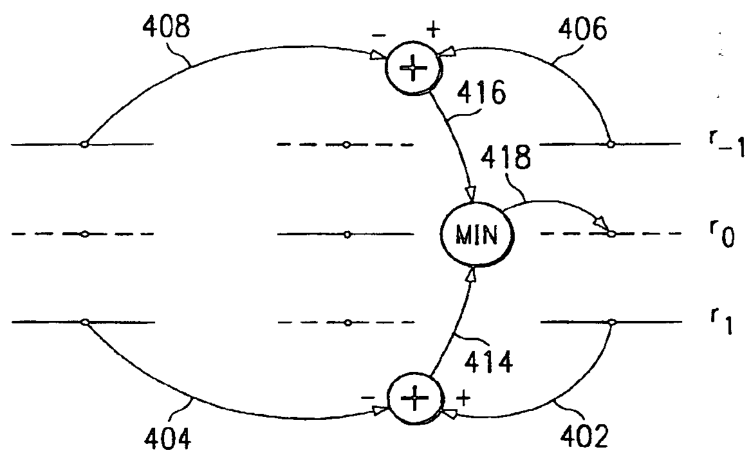


FIG. 5

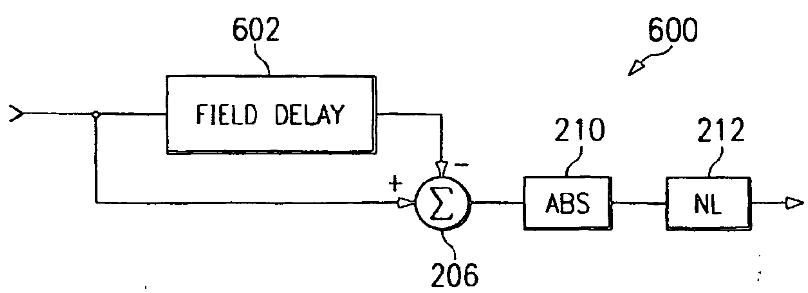


FIG. 6

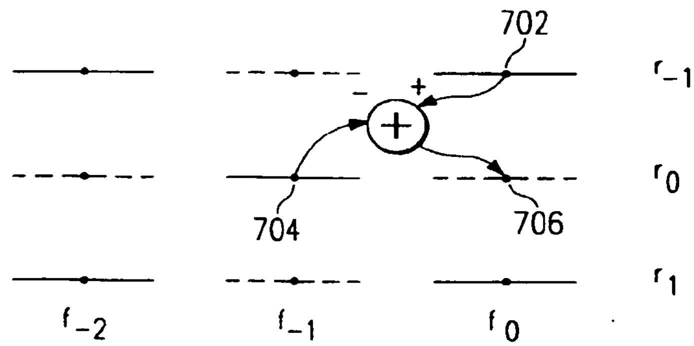

702
r-1
r0
r1
704
706
f-2
f-1
f0

FIG. 7

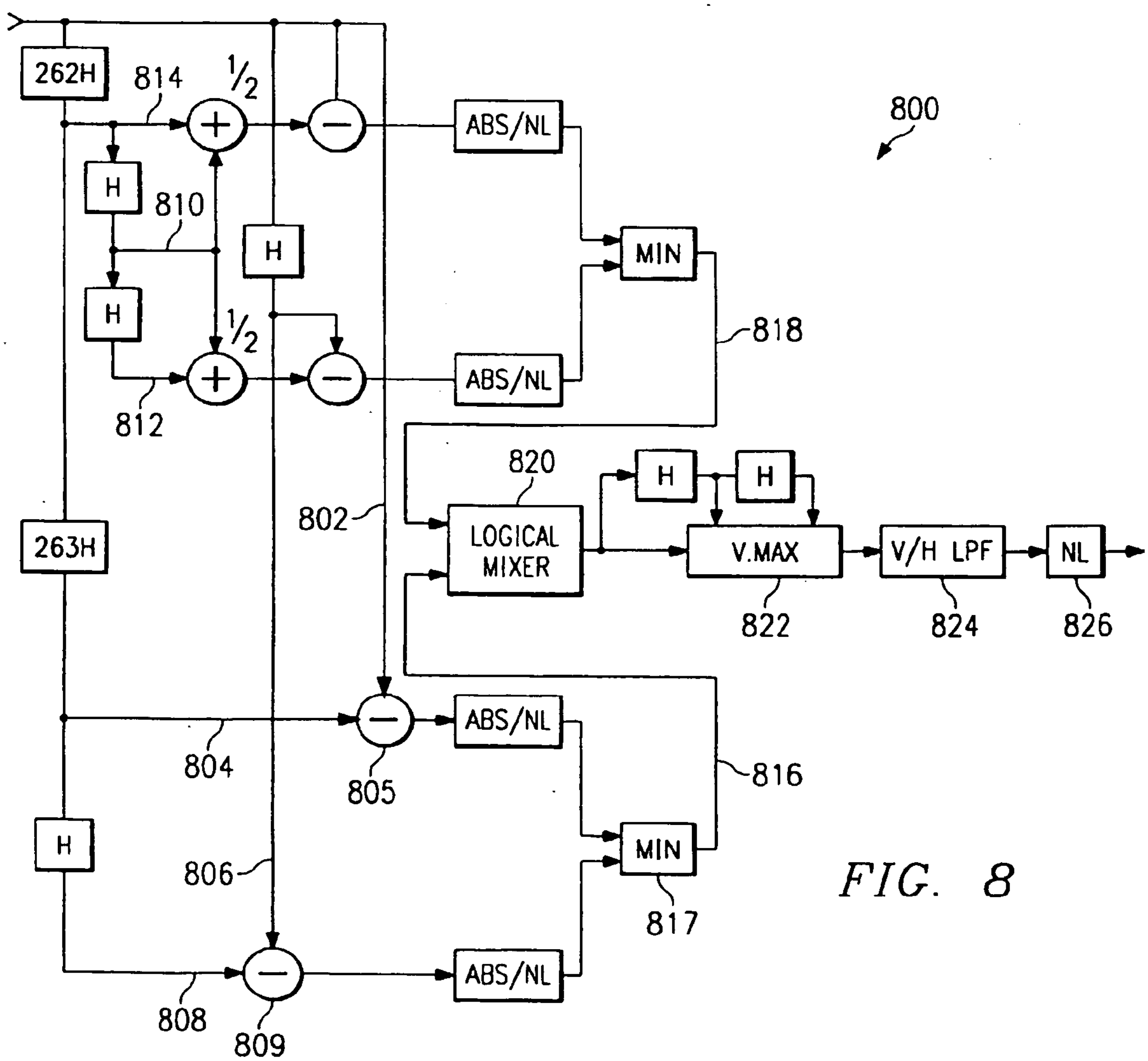
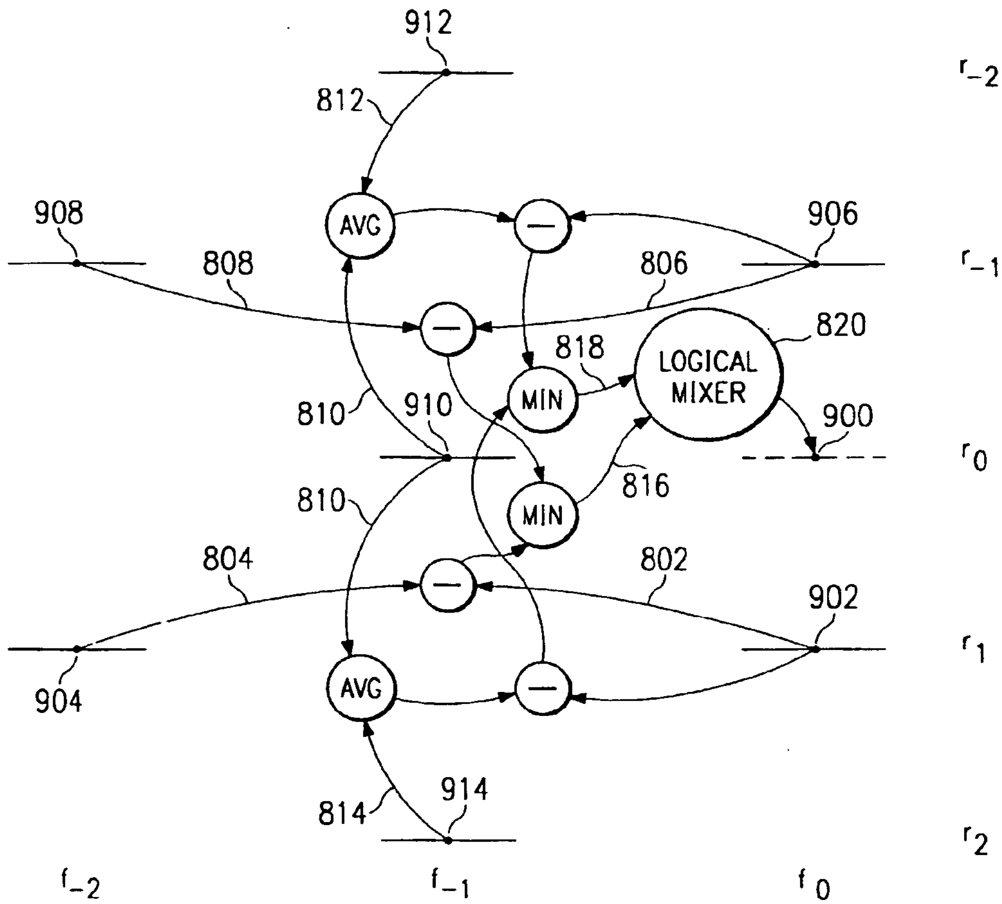
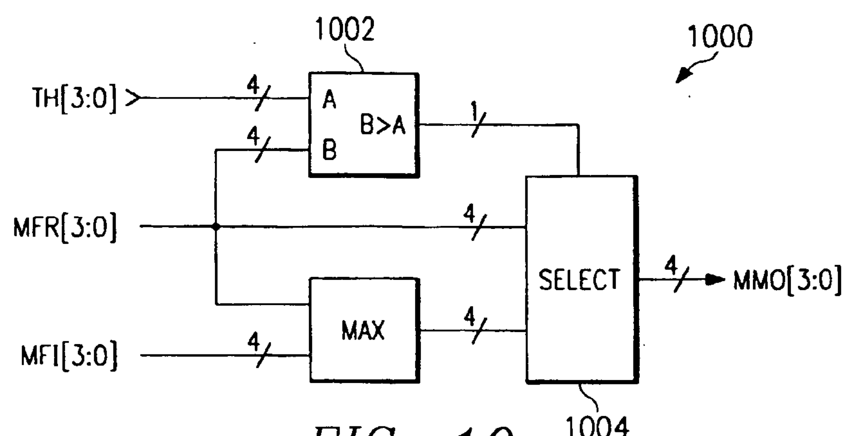

262H
814
1/2
810
H
H
H
812
1/2
ABS/NL
ABS/NL
MIN
818
800
820
LOGICAL MIXER
H
H
V.MAX
V/H LPF
NL
822
824
826
802
263H
804
805
ABS/NL
816
MIN
817
806
H
808
809
ABS/NL

FIG. 8



912
812
r_{-2}
908
AVG
906
r_{-1}
808
806
820
818
LOGICAL MIXER
810
910
MIN
900
r_0
810
816
MIN
804
802
902
r_1
904
AVG
914
814
r_2
f_{-2}
f_{-1}
f_0

FIG. 9



1002
1000
TH[3:0]
A
B>A
B
MFR[3:0]
SELECT
MMO[3:0]
MAX
MFI[3:0]
1004

FIG. 10

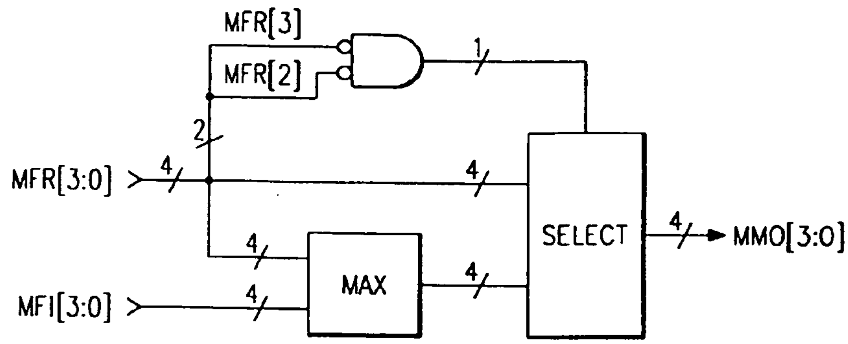


FIG. 11

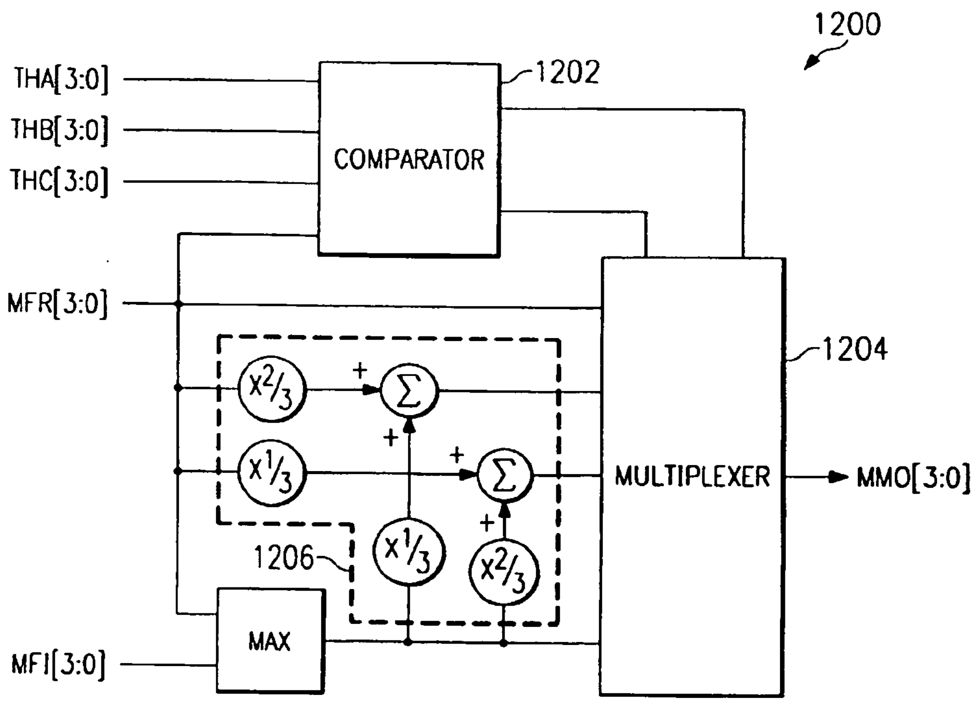


FIG. 12

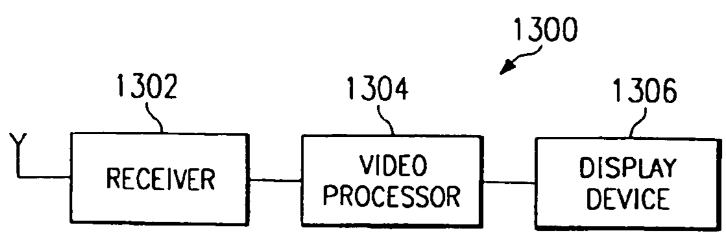


FIG. 13

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 830 018 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
29.12.1999 Bulletin 1999/52

(43) Date of publication A2:
18.03.1998 Bulletin 1998/12

(21) Application number: **97202808.8**

(22) Date of filing: **12.09.1997**

(51) Int Cl.6: **H04N 5/44**, H04N 5/14, H04N 7/36

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: **13.09.1996 US 26026 P**

(71) Applicant: **TEXAS INSTRUMENTS INCORPORATED**
Dallas, Texas 75243 (US)

(72) Inventor: **Ohara, Kazuhiro**
Richardson, Texas 75082 (US)

(74) Representative: **Holt, Michael**
Texas Instruments Limited,
P.O. Box 5069
Northampton NN4 7ZE (GB)

(54) **Method and system for motion detection in a video image**

(57) A method of measuring the motion in video image data for a pixel which uses both field-difference and frame-difference motion values to generate a motion value having increased accuracy. Image data (806) from the same pixel in a prior row of the same field (906) is compared to image data (808) from the same pixel in the prior row of the prior frame (908), and the absolute value of the difference is compared to the absolute value of the difference in image data (802) from the same pixel in a following row of the same field (902) and image data (804) from the same pixel in the following line of the prior frame (904). The minimum of these two values is the minimum frame-difference motion value which is input into a logical mixer. Also input into the logical mixer is the minimum field-difference motion value which may be determined by comparing data (802, 806) from the same pixel of an adjacent line of the same field (902, 906) with image data (810) from the same pixel of the same line of the prior field. The average of image data (810) from the same pixel of the same line of the prior field and image data (812, 814) from the same pixel of two rows prior or two rows after of the prior field (912, 914) may be used instead of image data (810) from the same pixel of the same line of the prior field alone, to increase the accuracy of the measurement.

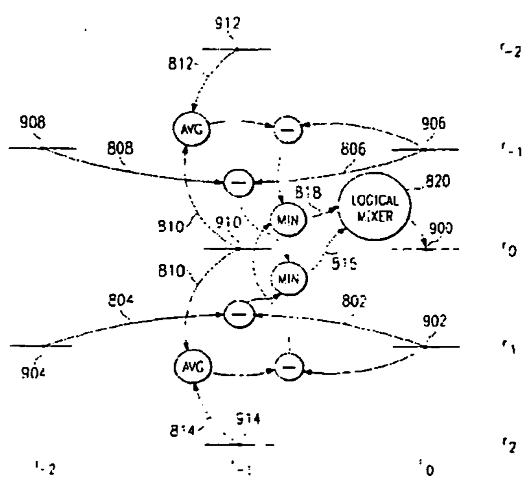


FIG. 9

EP 0 830 018 A3

Printed by Jouve, 75001 PARIS (FR)

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 97 20 2808

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	PATENT ABSTRACTS OF JAPAN vol. 015, no. 029 (E-1026), 23 January 1991 (1991-01-23) & JP 02 272984 A (NIPPON HOSO KYOKAI), 7 November 1990 (1990-11-07)	1,11	H04N5/44 H04N5/14 H04N7/36
A	* abstract *	2,3,12	
X	US 4 924 305 A (NAKAGAWA ISAO ET AL) 8 May 1990 (1990-05-08)	1,11	
A	* abstract; figures 1-3,13,14 *	2,3,12	
X	EP 0 688 133 A (EASTMAN KODAK CO) 20 December 1995 (1995-12-20) * column 3, line 35 - column 5, line 22; figure 7 *	1	
A	US 4 768 092 A (ISHIKAWA HISASHI) 30 August 1988 (1988-08-30) * column 4, line 46 - column 8, line 44; figures 3-9 *	1-3,12	
A	EP 0 662 767 A (TEXAS INSTRUMENTS INC) 12 July 1995 (1995-07-12) * page 2, line 15 - line 30 * * page 4, line 5 - page 6, line 23; figures 3-9 *	1	TECHNICAL FIELDS SEARCHED (Int.Cl.6) H04N
A	US 4 731 648 A (BERNARD FRANCIS S ET AL) 15 March 1988 (1988-03-15) * abstract *	1	

~~The present search report has been drawn up for all claims~~

Place of search	Date of completion of the search	Examiner
THE HAGUE	4 August 1999	FUCHS, P

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

European Patent
Office

Application Number
EP 97 20 2808

CLAIMS INCURRING FEES

The present European patent application comprised at the time of filing more than ten claims.

☐ Only part of the claims have been paid within the prescribed time limit. The present European search report has been drawn up for the first ten claims and for those claims for which claims fees have been paid, namely claim(s):

☐ No claims fees have been paid within the prescribed time limit. The present European search report has been drawn up for the first ten claims.

LACK OF UNITY OF INVENTION

The Search Division considers that the present European patent application does not comply with the requirements of unity of invention and relates to several inventions or groups of inventions, namely:

see sheet B

☐ All further search fees have been paid within the fixed time limit. The present European search report has been drawn up for all claims.

☐ As all searchable claims could be searched without effort justifying an additional fee, the Search Division did not invite payment of any additional fee.

☐ Only part of the further search fees have been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the inventions in respect of which search fees have been paid, namely claims:

☒ None of the further search fees have been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the invention first mentioned in the claims, namely claims:

1 - 15

European Patent Office

LACK OF UNITY OF INVENTION SHEET B

Application Number
EP 97 20 2808

The Search Division considers that the present European patent application does not comply with the requirements of unity of invention and relates to several inventions or groups of inventions, namely:

1. Claims: 1-15

   Method for converting interlaced video data to progressive scanned data.

2. Claims: 16-18

   Logical mixer

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 97 20 2808

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

04-08-1999

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
JP 02272984 A	07-11-1990	NONE	
US 4924305 A	08-05-1990	JP 1314493 A	19-12-1989
		JP 2634632 B	30-07-1997
		CA 1310401 A	17-11-1992
EP 0688133 A	20-12-1995	US 5473383 A	05-12-1995
		JP 8009341 A	12-01-1996
US 4768092 A	30-08-1988	JP 2770300 B	25-06-1998
		JP 63030077 A	08-02-1988
		JP 2104274 C	06-11-1996
		JP 8032023 B	27-03-1996
		JP 63030078 A	08-02-1988
		JP 2104275 C	06-11-1996
		JP 8032024 B	27-03-1996
		JP 63030079 A	08-02-1988
		JP 2775688 B	16-07-1998
		JP 63030080 A	08-02-1988
EP 0662767 A	12-07-1995	DE 69504373 D	08-10-1998
		DE 69504373 T	15-04-1999
		JP 8056336 A	27-02-1996
		US 5748250 A	05-05-1998
US 4731648 A	15-03-1988	DE 3789184 D	07-04-1994
		DE 3789184 T	08-09-1994
		EP 0262926 A	06-04-1988
		JP 2091521 C	18-09-1996
		JP 7121103 B	20-12-1995
		JP 63088980 A	20-04-1988
		KR 9509653 B	25-08-1995

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82